aws

COMMUNITY DAY

CAMEROON

22nd November 2025, Buea

aws COMMUNITY DAY CAMEROON

BUILDING A SERVERLESS DATA PIPELINE ON AWS:

A Case Study from Cameroon

aws COMMUNITY DAY CAMEROON

TENGU Quinter

AWS Solution Architect Instructor @ Trustech Online University

Founder & CEO: Elegancia Cakes & Treats

My Socials

LinkedIn: Tengu Quinter

Facebook: Quinter T Munah

Facebook: Elegancia Cakes and Treats

IG: Royal Elegancia Cakes

aws COMMUNITY DAY CAMEROON

Being guilty of missing my son's vaccination date—coupled with preventable tragedies like a neighbor's child catching measles and a friend's cancer linked to missed Hep B—exposed a vital need in my community. Many parents lack the literacy to manage complex medical records. The above motivated my project: an automated system that proactively reminds parents of their children's exact vaccination dates, ensuring no child misses a critical, life-saving shot.

aws COMMUNITY DAY CAMEROON

The Challenge

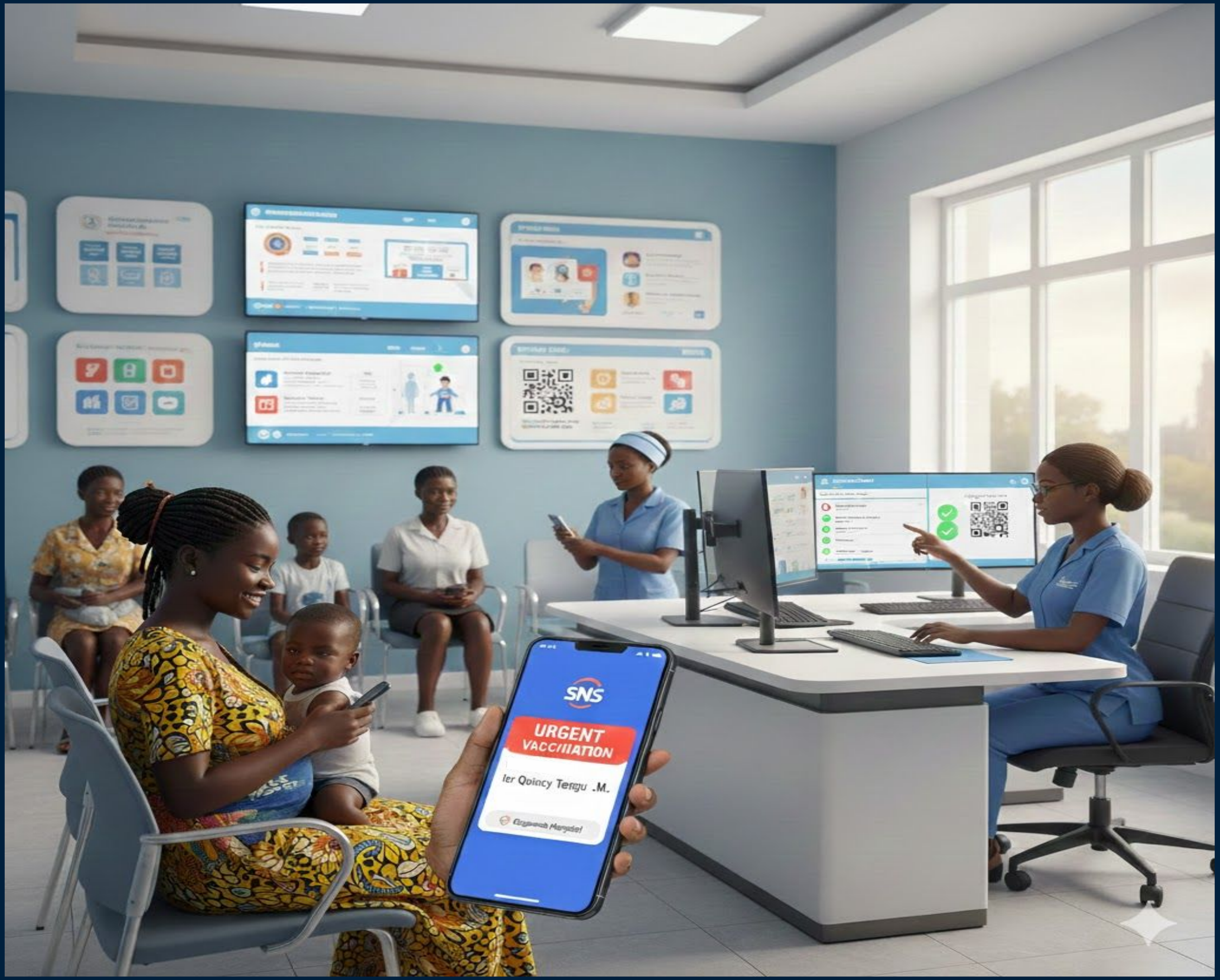
The main challenge is a **lack of systematic, timely communication** regarding childhood vaccination schedules in my community. This gap has lead to missed appointments, delays in critical immunizations, and a rise in preventable diseases, highlighting the need for a solution that overcomes illiteracy and logistical barriers.

The Challenge

Manual tracking of health records including vaccination records are prone to errors which can have life altering consequences.

aws COMMUNITY DAY CAMEROON

Problem Solved



The solution is an **automated, cloud-based reminder system** built on **AWS**. This system uses digital records to calculate each child’s upcoming due dates and proactively delivers personalized vaccination reminders via email (or SMS/call) to parents a week in advance, ensuring timely attendance.

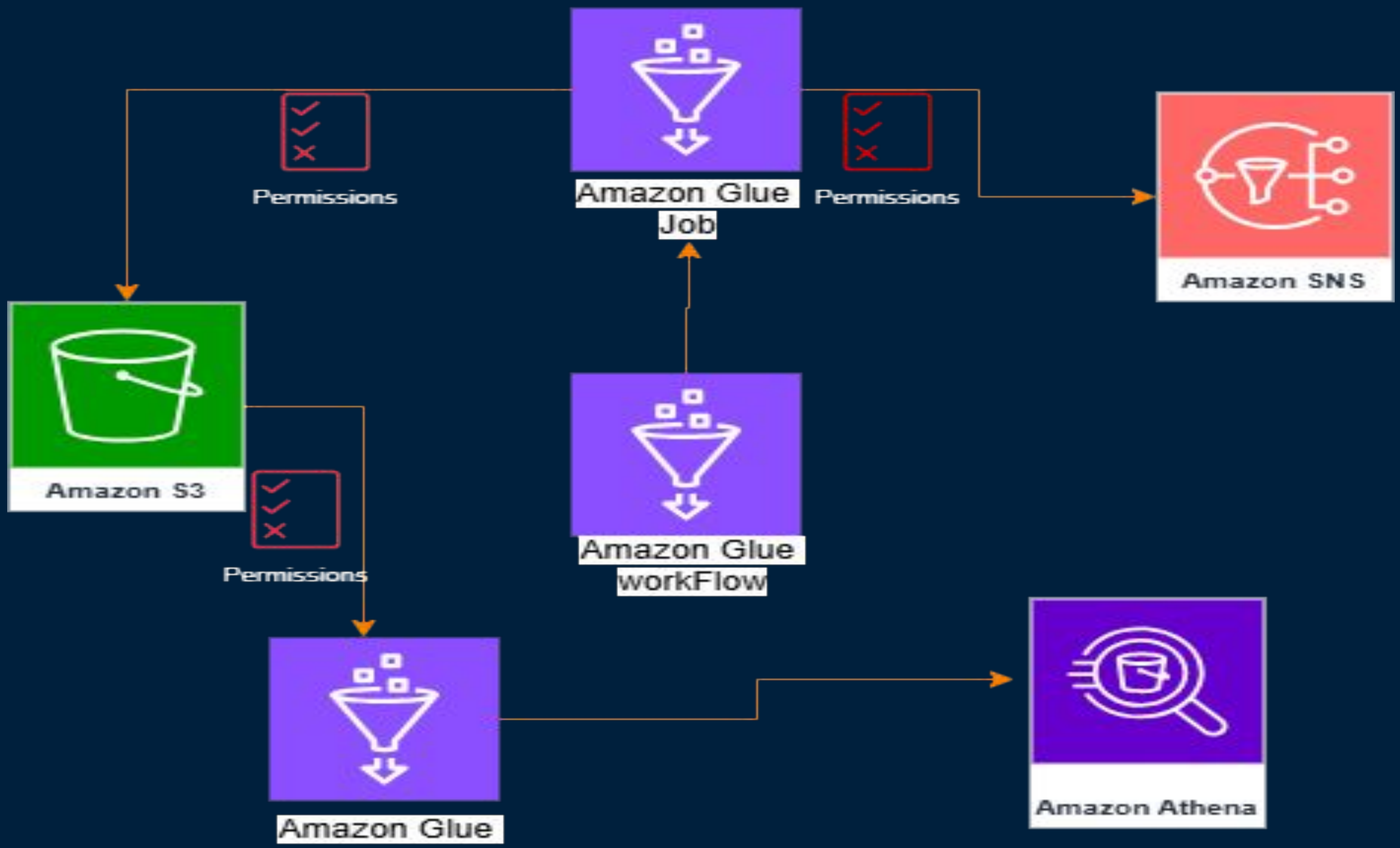
aws COMMUNITY DAY CAMEROON

Problem Solved

A Serverless Data Pipeline digitizes patient schedule, thereby replacing manual checks and ensuring proactive, timely, life-saving vaccination.

aws COMMUNITY DAY CAMEROON

A Serverless Data Pipeline on AWS



How It Works: the Pipeline Flow

1. **Data Ingestion:** A healthcare worker uploads a CSV file containing patient data to an S3 bucket.
2. **Automated Processing:** A **Glue Workflow** triggers a **Glue Job** on a daily schedule.
3. **ETL:** The Glue Job processes the data, identifying babies with near due vaccinations.

4. Proactive Notifications: The Glue Job publishes two separate messages to an **SNS topic**, one for babies ready for vaccination and another if there are no babies due for vaccination.

How It Works: the Pipeline Flow

aws COMMUNITY DAY CAMEROON

5. Email Reminders: SNS delivers these messages to the subscribed Public Relations Officer (PRO) via email.

6. Data Analysis: The PRO or other stakeholders can use **Athena** to run SQL queries or create a dashboard in **QuickSight** to analyze vaccination trends and forecasts.

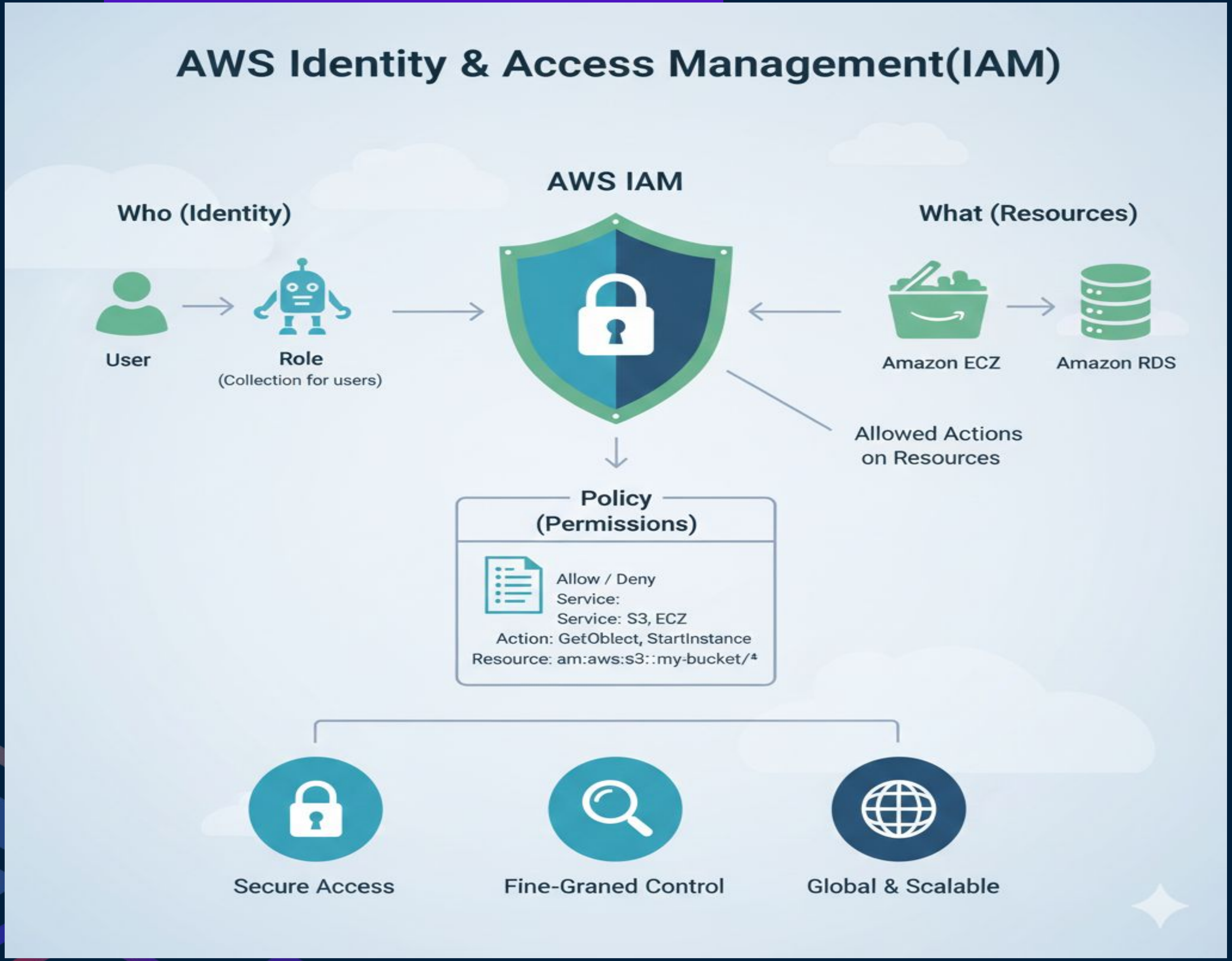
AWS Services: A Deep Dive

Upload / PUT
20 / AGT
CDN
Amazon S3
Access from anywhere
Scalable Storage in the Cloud
Durability & Availability
Security & Access Control
Global Reach & Data Tiering
S3 Standard
S3 Inferqurent Access
S3 Glacer

Amazon S3 (Simple Storage Service)

- Amazon S3 is a highly scalable and durable object storage service.
- **Role in Project:** acts as the project's **central, scalable data lake**. It reliably stores the CSV vaccination records, making the data accessible to **AWS Glue** for processing and **AWS Athena** for querying.

AWS Services: A Deep Dive

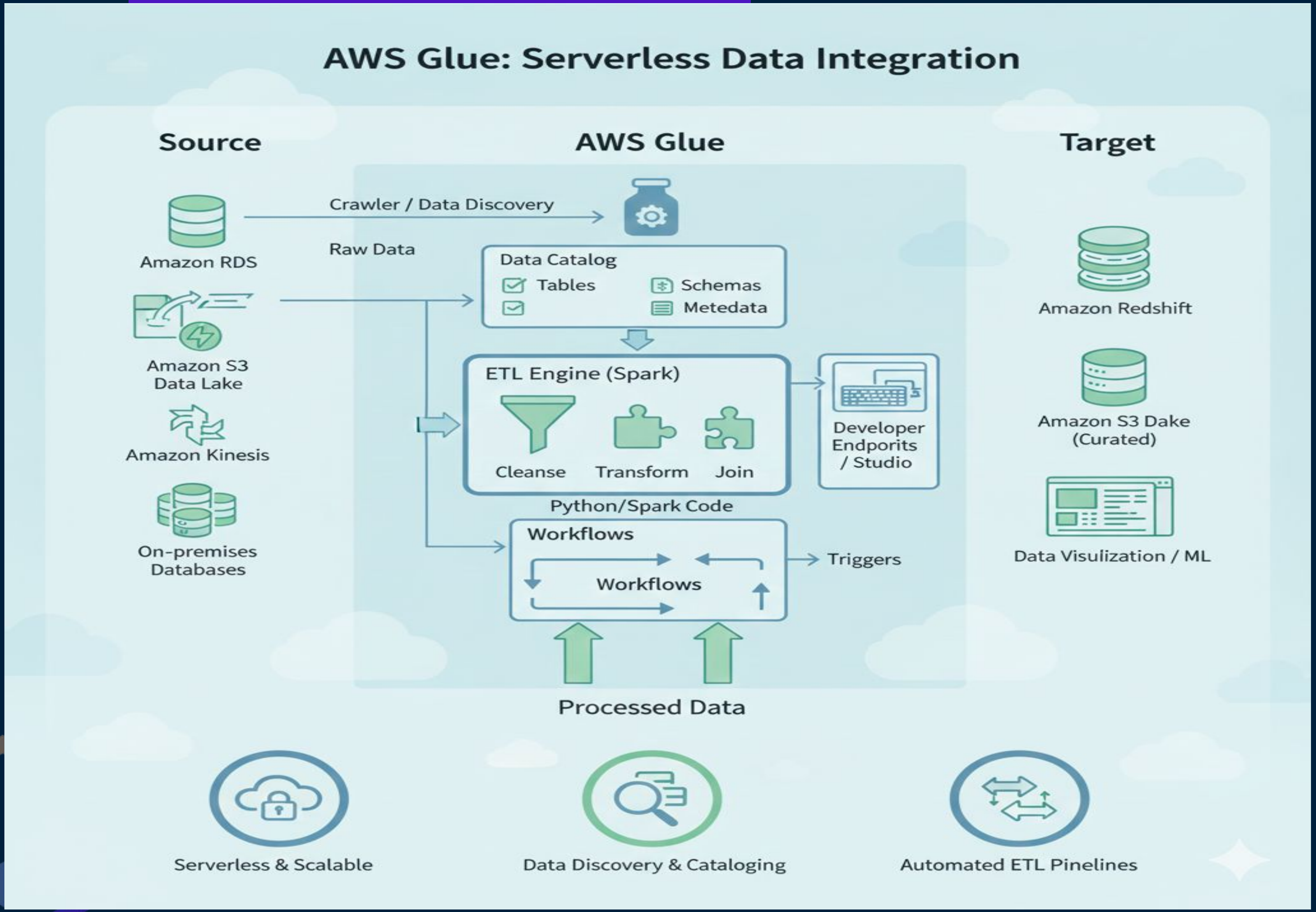


AWS IAM (Identity and Access Management

A service that enables you to manage access to AWS services and resources securely. It allows you to create and manage AWS users and groups, and use permissions to allow and deny their access to AWS resources.

Role in Project: The backbone of our project's security. We used it to create a specific role for the Glue job, granting it only the necessary permissions to read from S3 and publish messages via SNS. This ensures our pipeline is both functional and secure.

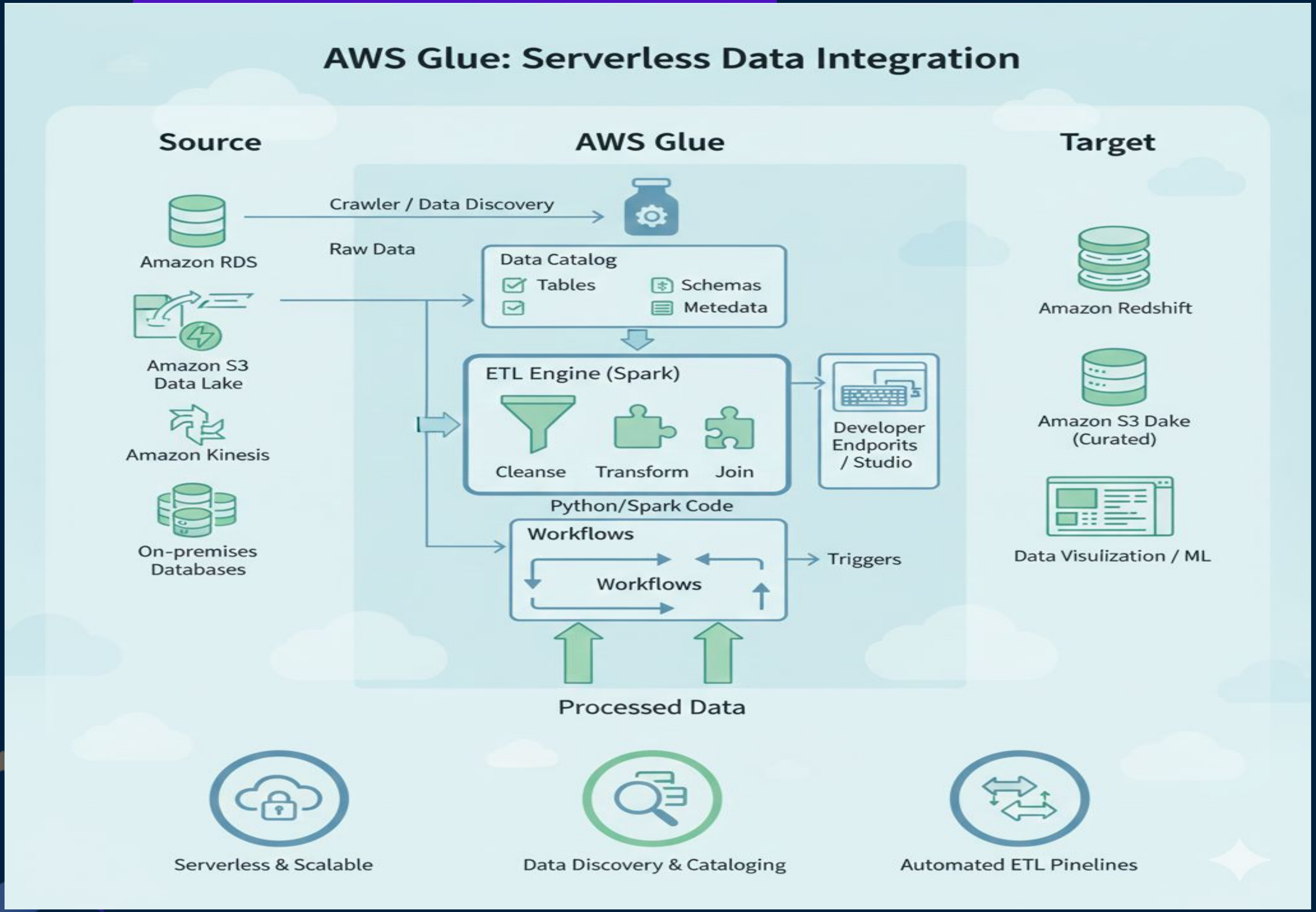
AWS Services: A Deep Dive



AWS Glue

- AWS Glue is a serverless data integration service that makes it easy to discover, prepare, and combine data.
- **Role in Project:** AWS Glue **runs the Python script (ETL job) for data transformation**. It reads vaccination data from S3, calculates due dates, determines reminders, and sends the final notifications via SNS.

AWS Services: A Deep Dive

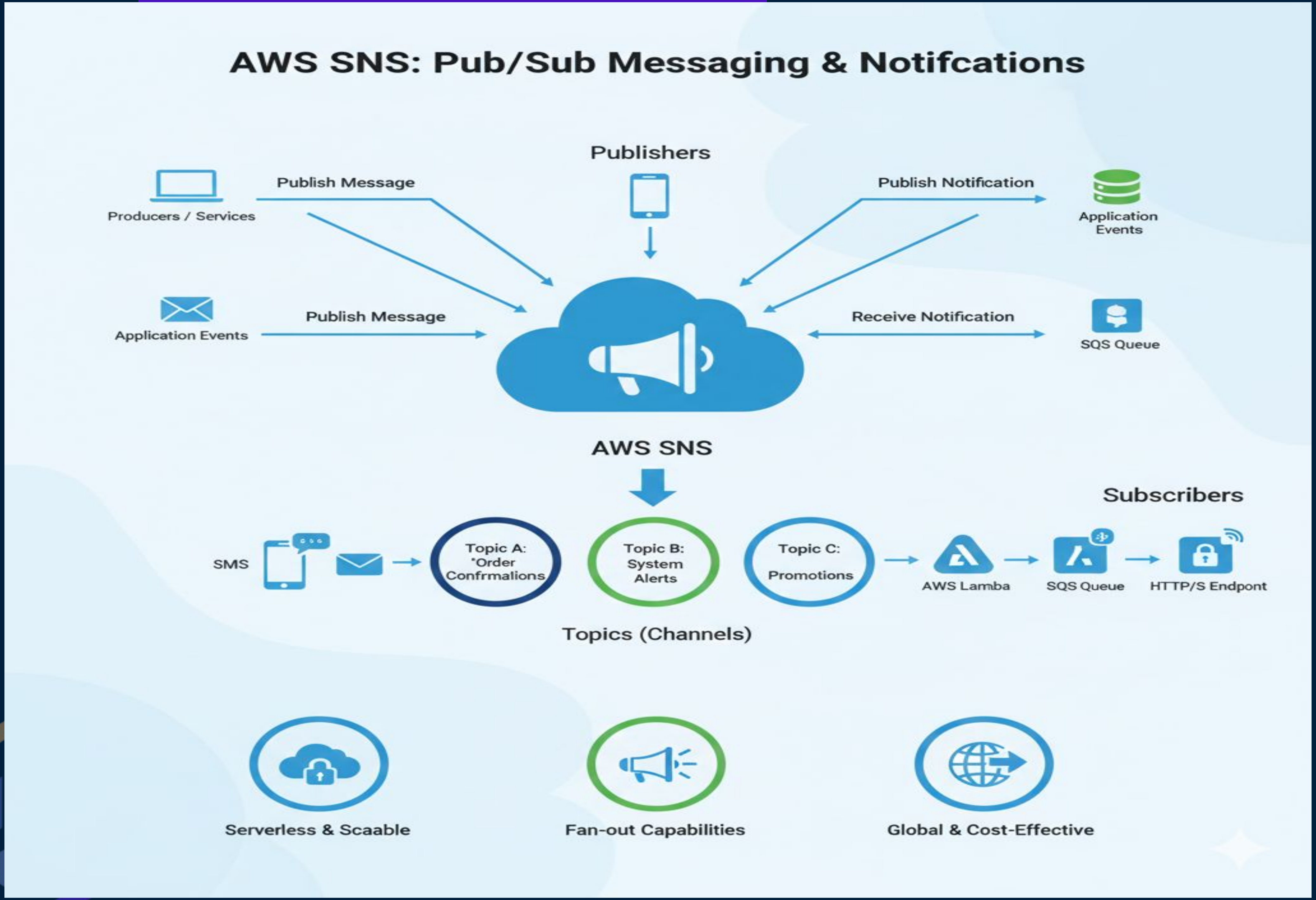


AWS Glue Workflow

- AWS Glue Workflow is a powerful orchestration service for creating and managing data pipelines.
- **Role in Project:** Acts as our scheduler and orchestrator. It manages the entire data pipeline, ensuring that the AWS Glue Job runs on a defined schedule and handles all dependencies in the workflow. This replaces the need for a separate EventBridge rule.

AWS Services: A Deep Dive

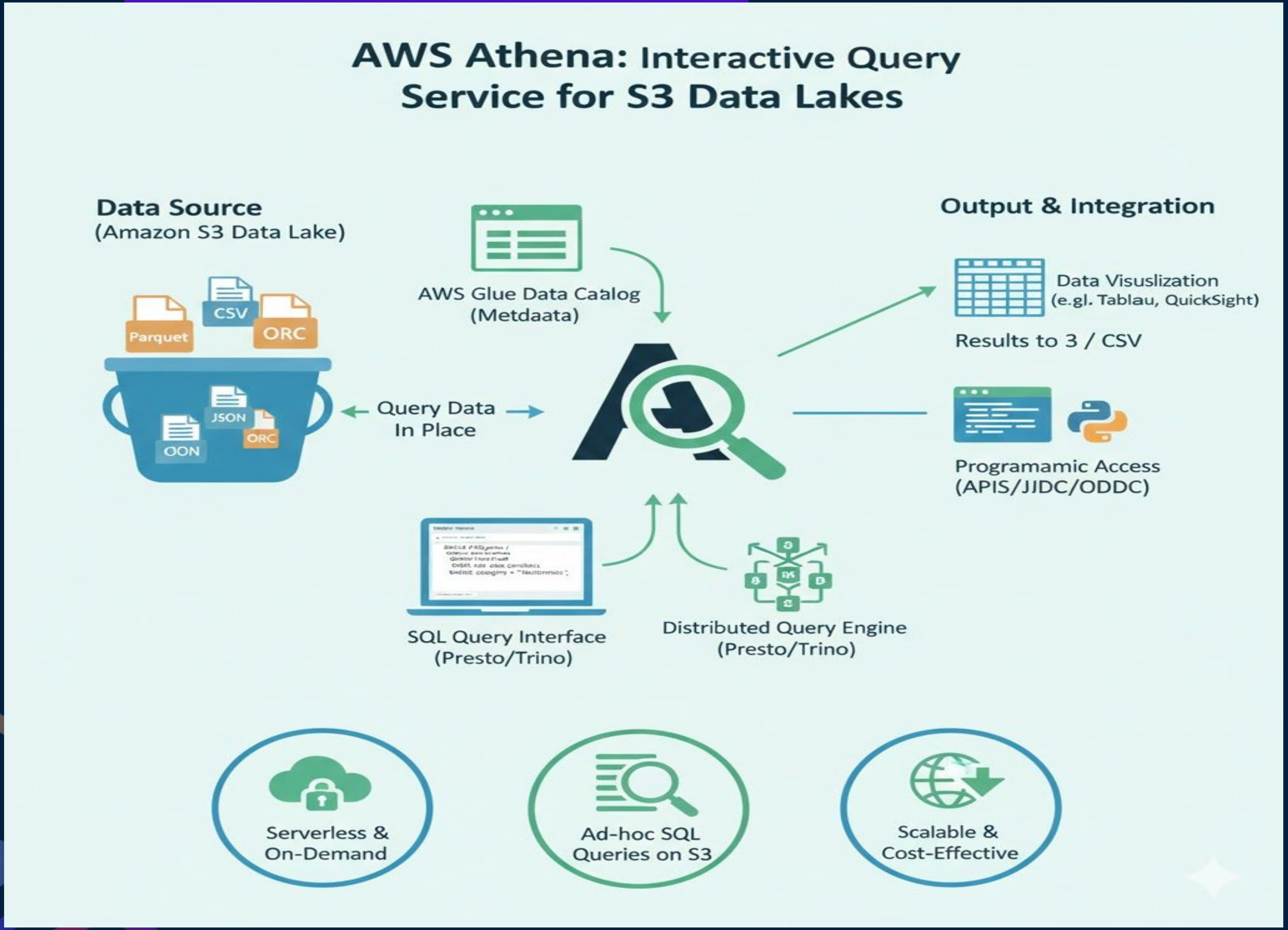
aws COMMUNITY DAY CAMEROON



Amazon SNS (Simple Notification Service)

- Amazon SNS is a fully managed messaging service for both application-to-application and application-to-person communication.
- **Role in Project:** Serves as the **decoupling broadcast mechanism.** Once the Glue Job identifies the reminders, it publishes a message to an SNS Topic. SNS then delivers this message to a subscriber, such as a Public Relations Officer via email.

AWS Services: A Deep Dive

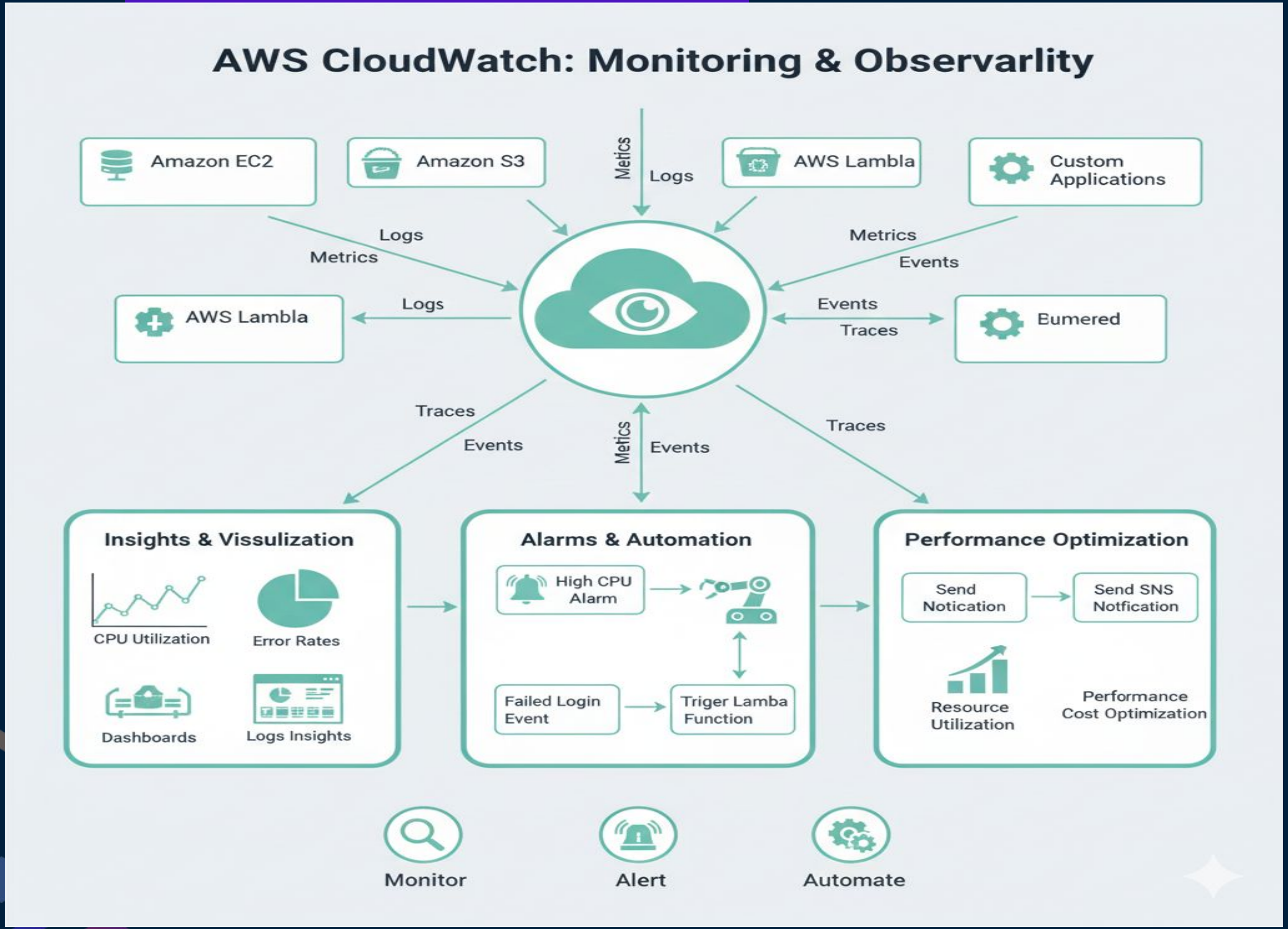

AWS Athena: Interactive Query Service for S3 Data Lakes
Data Source (Amazon S3 Data Lake)
Parquet
CSV
ORC
JSON
ORC
OON
AWS Glue Data Caalog (Metdaata)
Query Data In Place
Output & Integration
Data Visuslization (e.g|. Tablau, QuickSight)
Results to 3 / CSV
Programamic Access (APIS/JJDC/ODDC)
SQL Query Interface (Presto/Trino)
Distributed Query Engine (Presto/Trino)
Serverless & On-Demand
Ad-hoc SQL Queries on S3
Scalable & Cost-Effective

Amazon Athena

- Amazon Athena is a serverless query service that allows you to analyze data directly in Amazon S3 using standard SQL.
- **Role in Project:** Our interactive query service. We use Athena to run standard SQL queries directly against the data in S3. It allows us to perform powerful, on-demand analysis without having to load the data into a data warehouse.

AWS Services: A Deep Dive

aws COMMUNITY DAY CAMEROON



Amazon CloudWatch

A monitoring and observability service that provides you with data and actionable insights to monitor your applications, respond to system-wide performance changes, and optimize resource utilization.

Role in Project: Provides crucial monitoring and logging needed to troubleshoot and ensure our job runs successfully every time. We relied on its detailed logs to track the progress of our pipeline and debug any issues.

aws COMMUNITY DAY CAMEROON

SETUP AND INSTRUCTIONS

STEP 1: DATA AND STORAGE 📁

Preparing Our Dataset

Create Your Storage Folder (S3 Bucket)

- **Go to the Amazon S3 service** in your AWS account.
- Click the button to **"Create bucket."**
- Give it a unique, easy-to-remember name (like **vaccination-data-2025**). This name must be unique across all of AWS, like a web address.
- Leave all other settings as their default options and click **"Create bucket."**

SETUP AND INSTRUCTIONS

STEP 1: DATA AND STORAGE 📁

Preparing Our Dataset

Upload the File (Place the Records Inside)

- Click on the new bucket you just created (e.g., vaccination-data-2025) to open it.
- Click the **"Upload"** button.
- Drag and drop your hospital's **vaccination records CSV file** directly into this folder.
- Click **"Upload."** Your data is now securely stored and ready for the reminder system to use.

aws COMMUNITY DAY CAMEROON

STEP 2: NOTIFICATIONS 🔔

Setting up a Channel for Reminders

- **Create an SNS Topic (the Broadcast Channel):**
  The **Topic** is like creating a dedicated news channel or a bulletin board for one specific type of message—in this case, vaccination reminders.

1. **Go to the Amazon SNS service.**
2. Click **"Create topic."**
3. For the **Name**, use something clear like **pro-vaccination-reminders**.
4. For **Type**, choose **Standard**. (This is the normal, reliable type you need for reminders.)

AWS COMMUNITY DAY CAMEROON

STEP 2: NOTIFICATIONS 🔔

Setting up a Channel for Reminders

Add an Email Subscription (Sign Up the Recipient):

Once the channel exists, you need to sign up the Public Relations Officer (PRO) to receive the messages broadcast on that channel.

1. Inside your new pro-vaccination-reminders Topic, click **"Create subscription."**
2. For **Protocol**, select **Email.** (This tells the system how the message should be delivered.)
3. For **Endpoint**, type the PRO's exact email address (e.g., pro@example.com).

aws
COMMUNITY DAY
CAMEROON

STEP 2: NOTIFICATIONS 🔔

Setting up a Channel for Reminders

Confirm Subscription (Verify the Connection):

This is a security step to make sure you didn't accidentally sign up the wrong person.

1. The PRO must immediately **check their email inbox** for a message from AWS.
2. Inside that email, they must **click the confirmation link**.

Once they click the link, the subscription is active! Your automated system can now send a reminder to the channel, and the PRO's email will instantly receive it.

AWS COMMUNITY DAY CAMEROON

STEP 3: GRANT PERMISSIONS

Create IAM Role:

- **Go to the AWS IAM (Identity and Access Management) service.**
- In the left menu, click **"Roles,"** then click **"Create role."**
- For **Trusted entity type**, select **"AWS service."**
- For the use case, select **"Glue"** (this is the service that will assume this role).
- Click **"Next."**
- Give the role a clear, descriptive name (e.g., `Glue-Vaccination-Reminders-Role`).

aws COMMUNITY DAY CAMEROON

STEP 3: GRANT PERMISSIONS

Attach the Security Permissions (Policies)

You need to attach specific **Policies** (sets of rules) to this role so the Glue job can perform its required tasks:

Task	Policy Name to Attach	What it Grants Access To
Data Processing	AWSGlueServiceRole (Managed Policy)	Essential default permissions that allow Glue to run jobs, manage the Data Catalog, and read its own necessary files.a
Read Data	AmazonS3FullAccess	Allows Glue to read the vaccination records CSV file from your S3 bucket.
Send Alerts	AmazonSNSFullAccess	Allows Glue to publish the final reminder messages to your SNS Topic.
Logging	CloudWatchFullAccess	Allows Glue to write job logs (status, errors, completion time) to CloudWatch so you can monitor the job's success or troubleshoot errors.

aws COMMUNITY DAY CAMEROON

STEP 4: THE GLUE JOB 💻

The Engine of Our Pipeline

•Modify the Glue script:

Before you upload the script, you must replace the placeholder information in the Python code with the actual values you set up in the previous steps. This tells the script exactly where to find the data and where to send the messages.

- **S3 Bucket Name:** Find the name of your S3 storage folder (e.g., `vaccination-data-2025`).
- **S3 Key (File Name):** Find the exact name of the vaccination records file (e.g., `Elegancia District hospital Vaccination Records.csv`).
- **SNS Topic ARN:** Find the unique address for your reminder channel (the `pro-vaccination-reminders` Topic ARN, which starts with `arn:aws:sns...`).

You will paste these three values into the designated variables at the beginning of your Python script.

aws COMMUNITY DAY CAMEROON

STEP 4: THE GLUE JOB 💻

The Engine of Our Pipeline

Create the Glue Job

This step is about telling AWS where your worker lives and what tools it needs to do the job.

1. **Go to the AWS Glue service.**
2. In the left menu, select **"Jobs."**
3. Click **"Create job."**
4. For **"Script editor,"** choose the option to **"Create and edit script."**
5. For **Type,** make sure you select **"Spark"**. (Your Python script is written using PySpark libraries, and Spark is the engine that runs it quickly).
6. Uplaod script from your pc and create script.

aws COMMUNITY DAY CAMEROON

STEP 4: THE GLUE JOB 💻

- Go to **Job details** and entire the details.
- **Name:** vaccination-reminder-job.
- **IAM Role:** Select the `Glue-Vaccination-Reminders-Role` you created.

Language: python 3

- **Script Path:** s3://your-bucket/glue_script.py.
- **Worker Type:** G.1X.
- **Workers:** 2. we don't need the default 10 workers for this job.
- Leave other default configurations and save.
- You can test the job by running it manually.

aws COMMUNITY DAY CAMEROON

STEP 5: THE TRIGGER ⏰

The Daily Scheduler

This final step is like setting the **automated timer and sequence** for your data worker (the Glue Job). It makes sure your reminders are generated and sent out automatically, without you having to login and click "Run" every day.

We use an **AWS Glue Workflow** to handle this scheduling and sequencing.

- **Create Glue Workflow (Set Up the Automation Sequence):**
  - **Go to the AWS Glue service.**
  - In the left menu, select **"Workflows."**
  - Click **"Add workflow."**
  - Give it a clear name like `vaccination-reminder-workflow`.
  - Click **"Create."**

aws COMMUNITY DAY CAMEROON

STEP 5: THE TRIGGER ⏰

Create the Trigger (Set the Automatic Timer)

Next, you need to tell the workflow *when* to start running your reminder job. This is done by adding a **Trigger**—the alarm clock that starts the sequence.

1. Once inside your new workflow, click **"Add trigger."**
2. **Name the Trigger** (e.g., `daily-reminder-timer`).
3. For **Schedule Type**, choose **"Scheduled."**
4. For **Frequency**, set the timer:
   - **In testing:** Choose **Custom** and enter `rate(5 minutes` OR Cron Expression: 0/5 * * * ? *.) . (This makes the job run every 5 minutes so you can quickly see if it works.)
   - **In real life:** You would use a **Cron expression** like `0 6 * * ? *` to make it run once a day at 6:00 AM.
5. Click **"Add."**

aws COMMUNITY DAY CAMEROON

STEP 5: THE TRIGGER ⏰

Add Node (Connect the Timer to the Worker)

Now you connect the timer to the worker (your Glue Job).

1. Click **"Add node"** to connect your job to the workflow.
2. In the diagram, select the **Trigger** you just created.
3. Add the **Job** you created previously (`Daily-Vaccination-Reminder-Job`).

Start the Automation

1. Go back to the workflow view and click **"Start"** or **"Run"** on the main trigger.

That's it! The workflow will now automatically execute your `Daily-Vaccination-Reminder-Job` based on the schedule you set (e.g., every 5 minutes for testing), ensuring timely reminders are sent out.

aws
COMMUNITY DAY
CAMEROON

STEP 6: DATA ANALYSIS WITH ATHENA 📊

Insights from Our Data

This process is all about building a **digital catalog** so that our AWS search engine (Athena) can understand the vaccination data you stored in S3.

Think of the **Glue Crawler** as a **Librarian** you hire to automatically look at your raw CSV file in S3, figure out what columns it has (`Baby Name`, `Date of Birth`, etc.), and record that information in a central **Library Catalog (Glue Database)**.

🧑‍💻 1. Hire the Librarian (Create the Glue Crawler)

1. **Go to the AWS Glue service.**
2. In the left menu, click **"Crawlers."**
3. Click **"Create crawler."**
4. **Name it** clearly (e.g., `vaccination-records-crawler`).

aws COMMUNITY DAY CAMEROON

STEP 6: DATA ANALYSIS WITH ATHENA 📊

1. **Tell the Librarian where the data is**: Under **"Data sources,"** point it to the S3 bucket where you uploaded the CSV file (e.g., `s3://vaccination-data-2025/`).
2. **Give the Librarian its Access Pass:** For the **IAM Role**, select the **`Glue-Vaccination-Reminders-Role`** you created before (the one with permissions for S3, SNS, and Glue).
3. **Create the Library Catalog:** Under **"Output,"** create a new **Database** and name it something like **`vaccination-reminder-db`**. This is where the Librarian will store the table definition.
4. Refresh and add Database

Run the Librarian and Build the Catalog

1. Once the Crawler is created, click **"Run crawler."**
2. The Librarian will go to S3, read the CSV file, and finish its job.
3. When done, go to **AWS Glue > Tables**. You should see a new table listed (likely named something like `vaccination_records` or `vaccination_data_2025`). This table is the **Entry in the Catalog** that describes your CSV file.

aws COMMUNITY DAY CAMEROON

STEP 6: DATA ANALYSIS WITH ATHENA 📊

Search the Data (Using Athena)

1. **Open the Table:** Go back to the **AWS Glue > Tables** section, find the table the crawler created, click **"Action,"** and select **"View data."** This action automatically launches the Athena console.
2. **Set up Search Storage:** Athena will ask you where to save the results of your searches. Go to **Settings** and designate a new folder in your S3 bucket (e.g., `s3://vaccination-data-2025/query-results/`) as your query result location.
3. **Run your Query:** Use simple SQL to search the data. Athena uses the table the crawler created, allowing you to ask questions like finding specific children:

SQL

```
SELECT
  * FROM "AwsDataCatalog"."vaccination-reminder-db"."vaccination_records"
WHERE
  "baby_name" IN ('Agnes Beye', 'Bella Mballa', 'Samir Ngono');
```

- **Note:** Your column names might be `col0`, `col1`, etc., if the crawler didn't correctly identify the header row. If that happens, you may need to use **`"col0"`** instead of `"baby_name"` in your query.

TROUBLESHOOTING 💡

Common Issues and Solutions

- ModuleNotFoundError: **No module named 'pyspark'**
  - **Fix:** Ensure your Glue job **Type** is **Spark**, not Python Shell.
- TABLE_NOT_FOUND
  - **Fix 1:** Verify your Glue crawler is pointing to the correct S3 path (e.g., s3://bucket/, not s3://bucket/file.csv).
  - **Fix 2:** In Athena, ensure the database dropdown is set to the correct database (e.g., vaccine-db2).
  - **Fix 3:** Check for typos in the database or table name.

Delete all resources you created to save cost.

THANK YOU